

## Introducción a la programación

### Tarea #4

Elaborado por: Ricardo Artavia Solano

**PROBLEMA #1:** Obtener una lista con los números primos que contiene una lista brindada.

#### I. Entender el Problema

| <u>Incógnita:</u>  | <u>Datos disponibles:</u>   | <u>Restricción</u>  |
|--|---|---|
| <ul style="list-style-type: none"><li>Números primos en la lista dada.</li></ul> | <ul style="list-style-type: none"><li>Lista con números</li></ul> | <ul style="list-style-type: none"><li>La lista debe contener números y no debe estar vacía.</li></ul> |

#### II. Obtener el Plan

- Invocar función
- Verificar que su argumento sea una lista y que sus elementos sean números.
- Ejecutar una función auxiliar.
- Revisar si el primer elemento de la lista es primo.
- Si lo es, sumarlo a l
- a función auxiliar, pero quitar del argumento de dicha función el primer dígito de la lista.
- Si no es primo seguir recorriendo la lista.
- Cuando se han recorrido todos los elementos, se imprime la lista formada por los elementos primos sacados de la lista original.
- Terminar el proceso

##### Identificadores

- Lista de elementos numéricos.
- Primer elemento de la lista.
- Índices de la lista.
- Contador que funciona como divisor.

#### III. Aplicar el plan

1. Leer entrada ingresada.
2. Verificar que el dato ingresado sea de tipo lista.
3. Si el dato ingresado no es de tipo lista, imprime el mensaje: "Error: La entrada no es de tipo lista"
4. Si la entrada es válida, se activa al función secundaria.
5. Revisa si la lista en su argumento está vacía, si no lo está, se activa la función terciaria y devuelve True o False dependiendo de si el número es primo o no.
6. Cuando es True, se concatena el elemento analizado a función secundaria haciendo un "slicing"
7. Se realiza una recursión sobre la función secundaria hasta haber analizado todos los elementos de la lista original.
8. Cuando la lista está vacía, se imprime la lista construida al concatenar todos los elementos primos en una sola lista.
9. Termina el proceso.

#### IV. Revisar la solución

| VARIABLES  |                    | Corre<br>Línea | Comentario  |
|------------|--------------------|----------------|---|
| Lista      | Lista de<br>primos |                |   |
| [21,13,80] | [ ]                | 1              | Chequear que sea una lista válida con elemento tipo "int"   |
| [21,13,80] | [ ]                | 2              | Ejecuta función secundaria  |
| [21,13,80] | [ ]                | 3              | Revisa si la lista está vacía, de no ser así ejecuta la función terciaria pasándole el primer dígito de la lista.             |
| [21,13,80] | [ ]                | 4              | La función terciaria analiza el elemento y devuelve True si es primo o False si no lo es.                                     |
| [13,80]    | [13] + función     | 5              | Cuando devuelve True, se concatena el elemento analizado a la función secundaria. (Recursión 1)                               |
| [13,80]    |                    | 6              | Cuando devuelve False, se recorre la lista original.  |
| [ ]        | [13]               | 7              | Cuando la lista analizada está vacía, se concatenan los elementos primos en una sola nueva lista y se imprime en la pantalla. |
| [ ]        | [13]               | 8              | Termina Proceso   |

#### V. Resultado en pantalla

```
>>> numeros([21,13,80])  
[13]
```

#### Copia del código en estudio:

#Entradas: lista de números enteros

#Salidas: una lista con los números primos de la lista original

#Restricciones: La lista debe contener numeros enteros y no estar vacía

```
def numeros(lista):  
    if isinstance(lista,list):  
        return lista_primos(lista)  
    else:  
        return "Error, el valor debe ser una lista"
```

```
def lista_primos(lista):
```

```
if lista == []:  
    return lista  
elif primos(lista[0],2):  
    return [lista[0]] + lista_primos(lista[1:])  
else:  
    return lista_primos(lista[1:])
```

```
def primos(num, contador):  
    if num == 0 or num == 1 or num == 2 or num == 3:  
        return True
```

```
    elif contador == num:  
        return True  
    elif num % contador == 0:  
        return False  
    else:  
        return primos(num, contador + 1)
```